Web-Scale Knowledge Extraction from Semi-Structured Tables

Eric Crestan
Yahoo! Labs
701 First Avenue
Sunnyvale, CA 94089, USA
ecrestan@yahoo-inc.com

Patrick Pantel
Yahoo! Labs
701 First Avenue
Sunnyvale, CA 94089, USA
ppantel@yahoo-inc.com

ABSTRACT
A wealth of knowledge is encoded in the form of tables on the World Wide Web. We propose a classification algorithm and a rich feature set for automatically recognizing *layout* tables and *attribute/value* tables. We report the frequencies of these table types over a large analysis of the Web and propose open challenges for extracting from *attribute/value* tables semantic triples (knowledge). We then describe a solution to a key problem in extracting semantic triples: *protagonist detection*, i.e., finding the *subject* of the table that often is not present in the table itself. In 79% of our Web tables, our method finds the correct protagonist in its top three returned candidates.

Categories and Subject Descriptors
I.2.6 [**Artificial Intelligence**]: Learning – *knowledge acquisition*.

General Terms
Algorithms, Experimentation, Measurement.

Keywords
Information extraction, structured data, web tables, classification.

1. INTRODUCTION
Mining the vast amount of knowledge present in tables on the Web has the potential to power many applications such as query expansion [3] and textual advertising [6]. Recent efforts have focused on teasing apart tables consisting of relational information from those used strictly for multi-column layouts and formatting [6], and other efforts on extracting schemas and knowledge in the form of relational tuples [1][2][5].

Relational tables considered in this paper encode facts, or semantic triples of the form $\langle p, s, o \rangle$, where p is a predicate or relation, s is the subject of the predicate and o is its object. These tables are rendered in many different ways, and of interest in this work are a specific table type called ATTRIBUTE/VALUE. These tables list one or more attributes but they rarely contain the *subject* in the table proper. ATTRIBUTE/VALUE tables are often used as factual sheets about an entity, such as for the specifications of a digital camera model. For example, Figure 1 illustrates an ATTRIBUTE/VALUE table consisting of the *List Price* and *Price* of a movie, where the table does not contain the actual movie name. From this example, we'd like to extract semantic triples such as:

⟨*List Price*, *Angels & Demons*, *$36.95*⟩
⟨*Price*, *Angels & Demons*, *$22.99*⟩
⟨*You Save*, *Angels & Demons*, *$13.96*⟩

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

In ATTRIBUTE/VALUE tables, normally one column is devoted to the attribute names (mapping to predicates p) and another column to the values of the attributes (mapping to the objects o). The biggest challenge in extracting semantic triples from ATTRIBUTE/VALUE tables lies in the detection of the *subject* of the table. We call this open research problem *Protagonist Detection*[1].

We investigate a random sample of 5000 HTML tables over a large crawl of 1.2 billion high quality English pages on the Web. We further filter the tables according to the following criteria: a) minimum of 2 rows; b) minimum of 2 columns; and c) no cell with more than 100 characters in it. The result was 1.3 billion tables. For each table, we asked paid human editors to classify it as a *Layout* table (i.e., non-relational tables, such as formatting or navigational tables), as an ATTRIBUTE/VALUE table, or as OTHER.

Not surprisingly, the majority of tables are for layout purposes, a total of 58%. 16% were ATTRIBUTE/VALUE tables and the remaining 26% were classified as OTHER types of relational tables.

2. TABLE CLASSIFICATION
We adopt a *Gradient Boosted Decision Tree* classification model - GBDT [4], which consists of an ensemble of decision trees for the classes ATTRIBUTE/VALUE, LAYOUT, and OTHER, fitted in a forward step-wise manner to current residuals. The model is trained on our 5000 manually annotated random sample of tables, using the features described below.

Each (non-global) feature was extracted per row and per column for the two first rows and columns, as well as the last row and column. Features are grouped into three distinct classes:

Global Layout Features: Accounting for the structure of the table as a whole, they include the maximum number of rows for each column, the maximum number of columns for each row and the maximum cell content length in characters.

Layout Features: Layout features are applied per column and per row. They are solely based on the size of the cells and their variance. They include features such as the average cell length, the variance in cell length, and the ratio of cells in a column or a row generated by a colspan attribute.

Content Features: The following set of features focus on cell content. Two subdivisions can be distinguished based on whether the feature involves html tags or textual content. *Html features* include the ratio of distinct tags in the row/column, the ratio of cells containing table header <*th*> tags, the ratio of cells

[1] Beyond *protagonist detection*, a system must discover which columns and rows list *attributes* and *objects*, as well as normalize objects, find canonical forms for attributes and objects, and ultimately fuse triples across tables.

Table 1. Classification performance of TabEx compared with various baselines. P = Precision; R = Recall; F = F-score.

	Layout			Attribute/Value			Other		
	P	R	F	P	R	F	P	R	F
Global Features	0.558	0.737	0.648	0.390	0.267	0.329	0.309	0.217	0.263
Layout Features	0.738	0.781	0.759	0.643	0.661	0.652	0.523	0.451	0.487
Html Features	0.761	0.768	0.764	0.618	0.704	0.661	0.580	0.480	0.530
Lexical Features	0.766	0.802	0.784	0.777	0.721	0.749	0.614	0.553	0.584
TabEx	0.798	0.805	**0.801**	0.767	0.764	**0.766**	0.664	0.598	**0.631**

containing an anchor text, and the ratio of cells containing a font change. *Lexical features* include the ratio of distinct strings in the row/column, the ratio of cells ending with the colon character, the ratio of cells where the content is a number, and the ratio of cells containing a digit.

2.1 Experimental Analysis

20-fold cross-validation over our 5000 randomly sampled tables is used in order to compare the performance of our classifier with several baseline versions of it using the different feature families described above. For each system, we report 3 measures: *precision* (P), *recall* (R) and *F-measure* (F). The results reported in Table 1 are an average over the 20 runs for each table type. The overall TABEX accuracy was 75.2%.

From the results obtained using only the *Global Features*, the lack of modeling power is clearly exposed. Using only the *Layout Features* improves greatly over the simpler *Global Features*. ATTRIBUTE/VALUE tables benefit the most from *Lexical Features*. This observation follows the intuition that those tables contain knowledge offering in most of the cases certain regularity in its content. Finally, TABEX, our system using all the features performs the best overall in F-measure.

3. PROTAGONIST DETECTION

Extracting the *predicate* and *object* of semantic triples from ATTRIBUTE/VALUE tables is generally straigthforward[1]. Difficult however is recovering the often absent subject, which we call the task of *protagonist detection*. There are mainly three different places where the protagonist could be found: a) within the table (occasionally found in the table with a generic attribute such as *name* or *model*); b) within the document or the html <title> tag; and c) anchor texts pointing to the page. While table cells and anchor texts offer well defined boundaries for identifying protagonist candidates, the document body proposes fewer clues. There is however a series of html fields that could help in defining entity boundaries such as the headers and the font tags.

Our corpus consists of 200 manually annotated tables. For each table, an editor identified the valid set of protagonists among the content of the document or the anchor text pointing to it. None of the cases presented to the editors lacked a protagonist, highlighting that most often ATTRIBUTE/VALUE tables do indeed contain relational knowledge.

In order to identify all possible candidates, even if it is present in a paragraph of the document, we took an N-gram based approach. All possible 1 to 12-grams were extracted from the document and the anchor text (obtained from a commercial search engine's web link graph). For each N-gram, its frequency combined with its position (e.g. *anchor text, title, header, body, table, font…*) was used as features for our GBDT regression model. For some tables, as many as 1700 candidates were extracted.

We ran a 20-fold cross-validation experiment and present the results in Figure 2. Our system is labeled *ProIde* and it is compared against a simple baseline system that ranks the

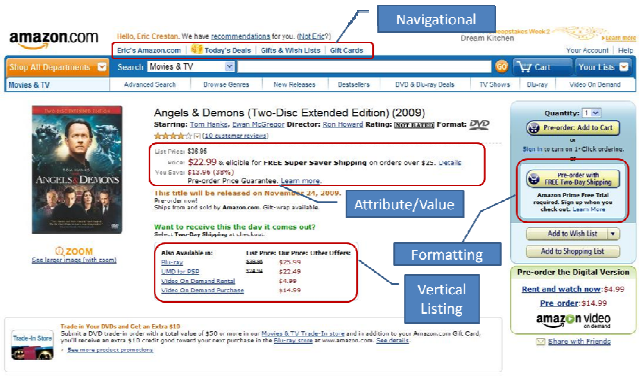

Figure 1. Example webpage containing multiple table types.

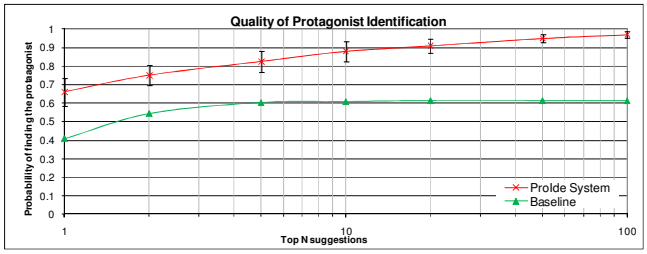

Figure 2. Probability of finding a correct protagonist vs. rank.

candidate protagonists according to their anchor text frequencies. This baseline achieves a surprisingly high precision of 40%. Although our system performs statistically significantly better than the baseline (by more than 25%), *ProIde* concedes 35% errors when looking at only the top suggestion and 12% errors when considering the top-10.

Our approach must be improved, but it is a good starting point for reducing the set of candidates in a first pass (97% chance to find the correct protagonist in the top-100 ranked candidates). Then, more expensive approaches could be used in order to verify whether the triples hold in other contexts using other extractors.

4. REFRENCES

[1] Cafarella, M.J.; Halevy, A.; Wang, D. Z.; Wu, E.; and Zhang, Y. 2008.. WebTables: Exploring the Powerpower of Tablestables on the Web. In *Proceedings of VLDB-08*. Auckland, New Zealand. pp. 538-549.

[2] Chen, H.; Tsai, S.; and Tsai, J. 2000. Mining Tables from Large-Scale HTML Texts. *In Proceedings of COLING-00*. Saarbrücken, Germany.

[3] J. H.; Jiang, D.; Pei, J.; He, Q.; Liao, Z.; Chen, E.; and Li, H. 2008. Context-aware query suggestion by mining click-through and session data. In *Proceedings of KDD-08*. pp. 875–883.

[4] Friedman, J.H. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232.

[5] Gatterbauer, W.; Bohunsky, P.; Herzog, M.; Krupl, B.; and Pollak, B. 2007. Towards Domain-Independent Information Extraction from Web Tables. *In Proceedings WWW-07*. pp. 71–80. Banff, Canada.

[6] Wang, Y. and Hu, J. 2002. A Machine Learning Based Approach for Table Detection on the Web. In Proceedings of WWW-02. Honolulu, Hawaii.